Evaluation & Design of Filters

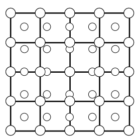
Torsten Möller, Simon Fraser Univ.
Raghu Machiraju, The Ohio State University

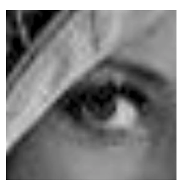
Other Significant Contributors:
Klaus Mueller, SUNY-Stony Brook
Roni Yagel, Insight Theraputics

Outline

- Motivation
- Theory
  - Approximation Theory
  - Signal Processing Approach
- Practical Considerations
  - Function Interpolation
  - Derivative Computation
- Summary

Image Processing





Nearest neighbor

Linear Interpolation

Practical Volume Rendering

- Numerical evaluation of Rendering Integral

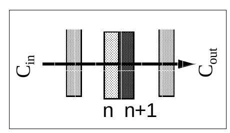
Discretization of rays necessary

Interpolate - $f(t)$

Determine opacity

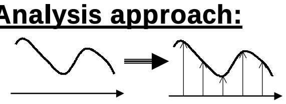
Determine shade

Compute gradient of $f(t)$



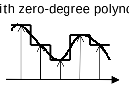
Framework

- **Numerical Analysis approach:**
- discretization: $f[k] = f(kT)$

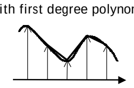


- reconstruction: Interpolation

Nearest neighbor
ppi with zero-degree polynomial



Linear interpolation
ppi with first degree polynomial



Problem Definition

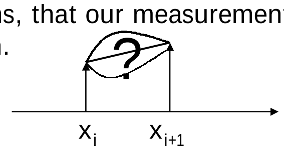
- Given a set of n+1 points $\{(x_i, f_i)\}$.
- Find a function f(x) defined for arbitrary x, such that

Approximation: $f(x_i) - f_i = \varepsilon$

Interpolation: $f(x_i) - f_i = 0$

Problem Definition (2)

- Generally there is **<u>no way</u>** to predict the values in-between the given (measured) points (x_i, f_i).
- We need knowledge of the space of functions, that our measurements were made in.



Mathematical Approach

- Usual assumptions are:
  - space of smooth functions C^n
  - space of bandlimited functions
- Function spaces characterized by specific basis functions $\phi_i(x)$,
- Each function expressed as a linear combination of these bases:

$$f(x) = a_0\phi_0(x) + a_1\phi_1(x) + \ldots$$

Mathematical Approach (2)

- different possible bases:
  - represent different function spaces
  - vary in quality
  - vary in computational efficiency
- Resort to:
  - Polynomial interpolation
  - Trigonometric functions
  - Wavelet basis

Polynomial Interpolation

- Given: Set of n+1 points S = {(x_i, f_i)}
- Sought: Representation of f(x) in the polynomial basis $\phi_i(x) = x^i$
- there is a unique polynomial p_n of degree n, that interpolates S

$$f(x) \approx p_n(x) = a_0 x^0 + a_1 x^1 + \ldots + a_n x^n$$

Lagrange Polynomials

- Defined as:

$$l_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}$$

  - very costly to evaluate - use
    - Power form or Horner's Method
    - Newtons form
  - accuracy problems
    - piecewise polynomials
    - orthogonal polynomials

Piecewise Polynomial Interp.

- Restrict to low degree polynomials that are “stitched” together.
- Many possible schemes for PPI, which differ in their accuracy, and features
  - Hermite,
  - Bezier
  - p^{th} degree Splines
  - B-splines/NURBS.

Orthogonal polynomials

- Computation of coefficients of $p_n(x)$ is unstable.
- Basis x^i is not orthogonal.
- Rewrite polynomial $p_n(x)$ in a different, orthogonal, basis P_i:

$$p_n(x) = b_0 P_0(x) + b_1 P_1(x) + \ldots + b_n P_n(x)$$

- with condition:

$$\langle P_i, P_j \rangle_w = \int_a^b P_i(x) P_j(x) w(x) dx = 0$$

Theory-Math Theory-SP Interpolation Derivative

Orthogonal polynomials (2)

- Common orthogonal polynomials:
  - Legendre
  - Chebychev
  - Trigonometric functions (not polynomials)
  - Bessel functions (not polynomials).

Theory-Math Theory-SP Interpolation Derivative

Trigonometric functions

- More than just a tool for interpolation
- Widely used for analysis of signals

Cont. Basis functions:

$$\phi_\omega(x) = e^{i\omega x}$$

Fourier Transform

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{iux} du$$

$$F(u) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-iux} dx$$

Discrete Basis functions:

$$\phi_j^n(k) = e^{i2\pi jk/n}$$

Discrete Fourier Transform

$$f_j = \sum_{k=0}^{n-1} F_k e^{i2\pi jk/n}$$

$$F_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{-i2\pi jk/n}$$

Theory-Math Theory-SP Interpolation Derivative

Fourier Transforms

There are 4 major transforms used

Name	Spatial D.	Freq. D.	Use
FT - Fourier Transform	cont.	cont.	for analysis
FS - Fourier Series	cont.	discret	Interpolation trad. Math appr.
DTFT - Discrete Time FT	discret	cont.	Filter Design trad. SP appr.
DFT - Discrete FT	discret	discret	Implementation (computer)

Theory-Math Theory-SP Interpolation Derivative

Requirements

- Performance
- Stability of the numerical algorithm
- Accuracy (numerical + perceptual)

Theory-Math Theory-SP Interpolation Derivative

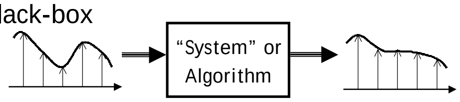
Requirements (2)

- Accuracy considerations depend on underlying function space.
- For smooth function spaces, we consider asymptotic error behaviors.
- For bandlimited spaces, we consider blurring, aliasing and overshoot (Frequency domain).
- Not considered – Perceptual metrics

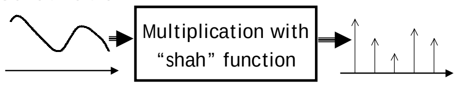
Signal Processing Ideas

Engineering approach:

- black-box



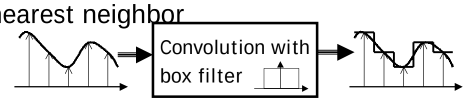
- discretization:



Signal Processing Ideas (2)

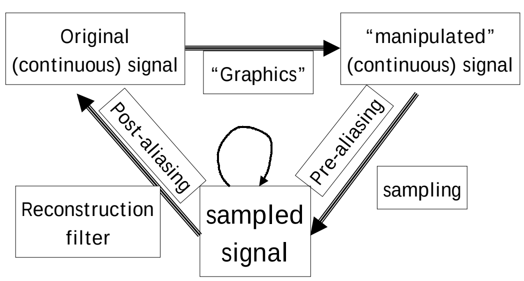
Engineering approach (cont.):

- nearest neighbor



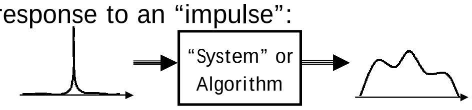
- linear filter:

Signal Processing Ideas (3)



Convolution

- How can we characterize our “black box”?
- We assume to have a “nice” box/algorithm:
  - linear
  - time-invariant
- then it can be characterized through the response to an “impulse”:



Convolution (2)

- Finite Impulse Response (FIR) vs.
- Infinite Impulse Response (IIR)
- Impulse:

$$\delta(x) = 0, \text{ if } x \neq 0$$

$$\int_{-\infty}^{\infty} \delta(x)dx = 1$$

- discrete impulse:

$$\delta[k] = 0, \text{ if } k \neq 0$$

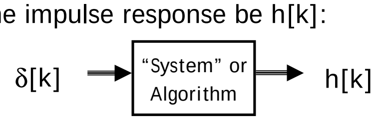
$$\delta[0] = 1$$

Convolution (3)

An arbitrary signal x[k] can be written as:

$$x[k] = \cdots + x[-1]\delta[k+1] + x[0]\delta[k] + x[1]\delta[k-1] + \cdots$$

Let the impulse response be h[k]:



Convolution (4)

- for a time-invariant system h[k-n] would be the impulse response to a delayed impulse δ[k-n]
- hence, if y[k] is the response of our system to the input x[k] (and we assume a linear system): $y[k] = \sum_{n=-N}^{N} x[n]h[k-n]$

IIR - N=inf.
FIR - N<inf.

x[k] → "System" or Algorithm → y[k]

Fourier Transforms

- Let's look at a special input sequence:

$$x[k] = e^{i\omega k}$$

- then:

$$y[k] = \sum_{n=-N}^{N} e^{i\omega(k-n)} h[n]$$
$$= e^{i\omega k} \sum_{n=-N}^{N} e^{-i\omega n} h[n]$$
$$= H(\omega) e^{i\omega k}$$

Fourier Transforms (2)

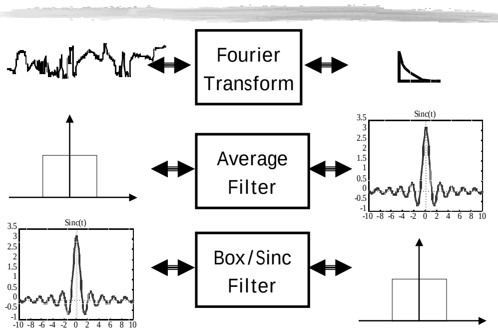
- Hence $e^{i\omega k}$ is an eigen-function and H(ω) its eigenvalue
- H(ω) is the Fourier-Transform of the h[n] and hence characterizes the underlying system in terms of frequencies
- H(ω) is periodic with period 2π
- H(ω) is decomposed into
  - phase (angle) response $\sphericalangle H(\omega)$
  - magnitude response $|H(\omega)|$

Fourier Transforms - Example

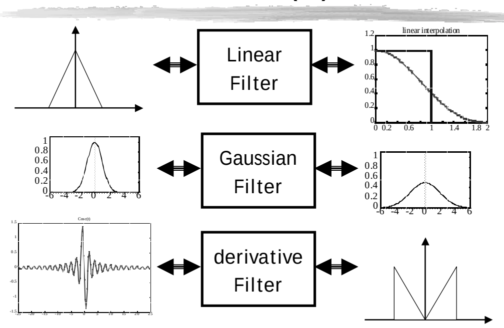
- Let's look at a simple example of averaging: $h[0] = h[1] = 0.5$
- then: $y[k] = \sum_{n=0}^{1} e^{i\omega(k-n)} h[n] = e^{i\omega k} \sum_{n=0}^{1} e^{-i\omega n} 0.5$

$$= e^{i\omega k} 0.5(e^{0} + e^{-i\omega})$$
$$= e^{i\omega k} 0.5 e^{-i\omega/2}(e^{i\omega/2} + e^{-i\omega/2})$$
$$= e^{i\omega k} e^{-i\omega/2} \cos \omega/2$$

Transforms Pairs



Transforms Pairs (2)



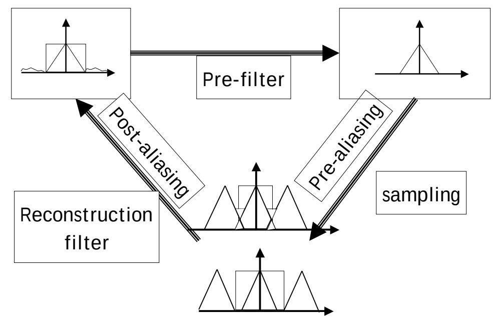
Transform Pairs - Shah

- Sampling = Multiplication with a Shah function:

- multiplication in spatial domain = convolution in the frequency domain
  - frequence replica of primary spectrum (also called *aliased* spectra)

Theory-Math Theory-SP Interpolation Derivative

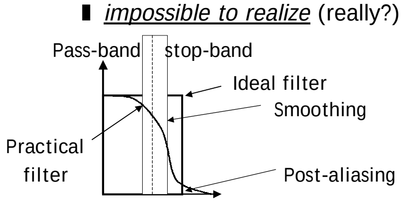
Basic Pipeline



Theory-Math Theory-SP Interpolation Derivative

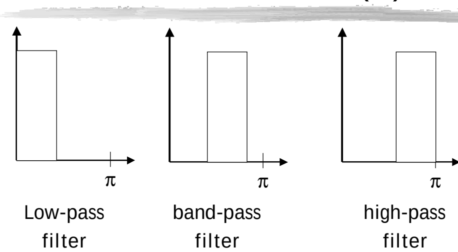
Ideal Reconstruction

- Box filter in frequency domain =
- Sinc Filter in spatial domain
  - *impossible to realize* (really?)



Theory-Math Theory-SP Interpolation Derivative

Ideal Reconstruction (2)



Realizable filters do not have sharp transitions; also have ringing in pass/stop bands

Theory-Math Theory-SP Interpolation Derivative

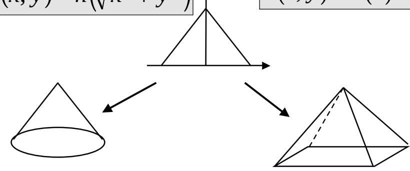
Higher Dimensions

- An-isotropic Filters
- (radially symmetric)

$$h(x,y)=h\left(\sqrt{x^2+y^2}\right)$$

separable filters

$$h(x,y)=h(x)\cdot h(y)$$



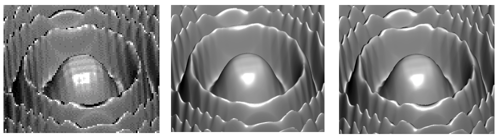
Theory-Math Theory-SP Interpolation Derivative

Frequency Domain Issues

- Postaliasing
  - reconstruction filter passes frequencies beyond the Nyquist frequency (of duplicated frequency spectrum)
  - Frequency components of the original signal appear in the reconstructed signal at different frequencies
- Smoothing
  - frequencies below the Nyquist frequency are attenuated

Frequency Domain Issues (2)

- Ringing (overshoot)
  - occurs when trying to sample/reconstruct discontinuity
- Anisotropy
  - caused by not spherically symmetric filters



Aliasing OK Blurring

Filter Evaluation/Design

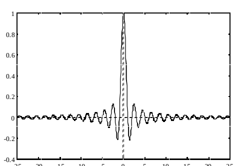
Spatial Domain	Frequency Domain
local features	global features
minimal error	robust noise handling
smoothness	no blurring
CAD, numerical solvers	no aliasing
	DSP chips (audio, video)
➲ Approximation Theory/Analysis	➲ Signal Processing
discrete moments	DFT/Laplace Transform
asymptotic error	divergence from Ideal

Ideal Interpolation

Spatial Domain:

convolution is exact

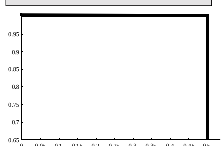
$$\boxed{f_r(x) - f(x) = 0}$$



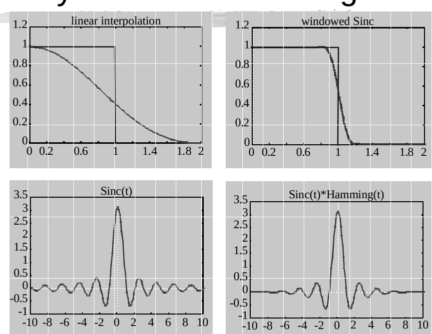
Frequency Domain:

cut off freq. replica

$$\boxed{\mathrm{Sinc}(x) = \frac{\sin(\pi x)}{\pi x}}$$



An easy fix - Windowing



An easy fix - Windowing

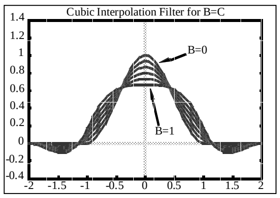
- Multiply Sinc with a window:
  - Rectangular $wind(t) = \begin{cases} 1 & -M \le t \le M \\ 0 & \text{otherwise} \end{cases}$
  - Bartlett $wind(t) = \begin{cases} 1 + t/M & -M \le t \le 0 \\ 1 - t/M & 0 \le t \le M \\ 0 & \text{otherwise} \end{cases}$
  - Hamming/Hanning

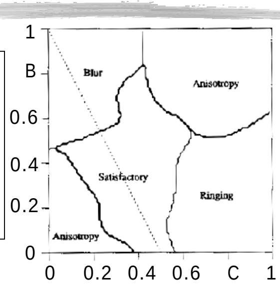
$$wind(t) = \begin{cases} \alpha + \beta \cos\left(\pi \frac{t}{M}\right) & -M \le t \le M \\ 0 & \text{otherwise} \end{cases}$$

Spatial Design

- Keys - Cardinal Splines
  - cubic interpolation filter - symmetric & C^2
  - compares methods in spatial & frequency d.
- Mitchell/Netravali - BC-splines
  - cubic interpolation filters - symmetric & C^1
  - cardinal splines are subset
  - numerical & user study

Mitchell & Netravali





Möller, Machiraju, Mueller, Yagel

- Assumes smooth function space
- General scheme for spatial accuracy evaluation of filter functions
- Generalization of Keys' method using a Taylor series expansion
- Enables filter evaluation and design

Accuracy

- Taylor Series:
  - approximate the error of the numerical algorithm
  - evaluate its asymptotic behavior.
- Assumption:

  some derivatives of the underlying function *f* exist.

$$f(x) = f(t) + f'(t)\frac{(x-t)}{1!} + f''(t)\frac{(x-t)^2}{2!} + \ldots$$

Spatial Domain Evaluation

- Reconstruction of the underlying function:

$$f_r(t) = a_0 f + \ldots + a_{N-1} f^{(N-1)} + a_N f^{(N)} + a_{N+1} f^{(N+1)} + \ldots$$

Normalize	Is $a_0(\tau) = 1$?
Classify	How many $a_l(\tau)=0$ for $l>N$?
Error	How large is the error term?

Meaning of the a's

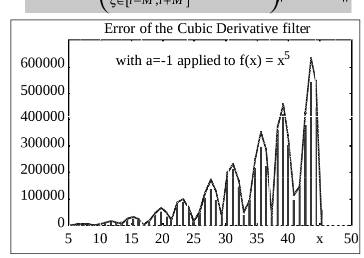
$$f_r(t) = a_0 f + \ldots + a_{N-1} f^{(N-1)} + a_N f^{(N)} + a_{N+1} f^{(N+1)} + \ldots$$

$$a_n^w(\tau) = \frac{1}{n!}\sum_{k=-\infty}^{\infty}(k-\tau)^n w(\tau-k)$$

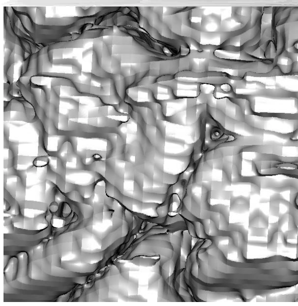
- They are nothing but discrete moments !
- And one can also prescribe the accuracy of reconstructed function

Error Estimation

$$r_{L,i}^w(\tau) \le \left(\max_{\xi\in[i-M,i+M]} f^{(L+1)}(\xi)\right)\left\|a_{L+1}^w(\tau)\right\|$$



Accuracy Not Enough



- Missing perceptual influence

Smoothness

- Smoothness = Visual Artifacts
- Reconstructed Function:

$$f_r(t) = \ldots + f(k-1)w(t-(k-1)) + f(k)w(t-k) + \ldots$$

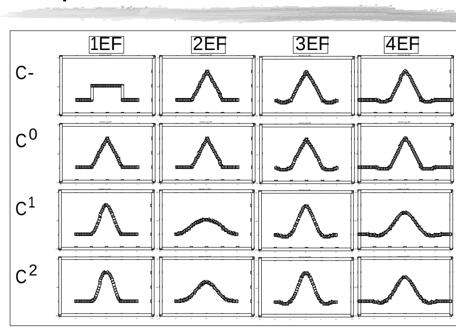
For f to be smooth (in C^M) the filter weights must be smooth:

$$w_k \in C^M \qquad w_k^{(m)}(1) = w_{k+1}^{(m)}(0)$$

Spatial Filter Design

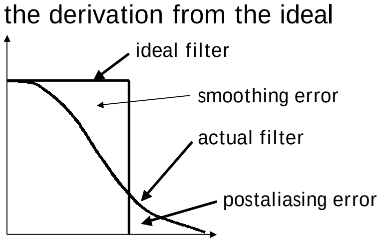
What **ACCURACY** do we require from the reconstruction process? ⟷ How **SMOOTH** (space C^n) should the reconstructed function be?

Interpolation filters

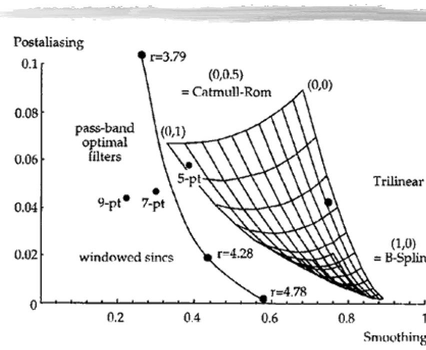


Marschner/Lobb

- Frequency Domain Design
- Measure the derivation from the ideal



Marschner/Lobb Evaluation



Theory-Math | Theory-SP | **Interpolation** | Derivative

Carlbom

- Approximates ideal frequency response
- Uses Remez algorithm to minimize error:
  $E(\omega) = \max(W(\omega) \mid H(\omega) - F(\omega) \mid)$
- With ideal interpolation filter

$$F(\omega) = e^{-i2\pi\omega\tau}$$

- Where τ is the reconstruction offset

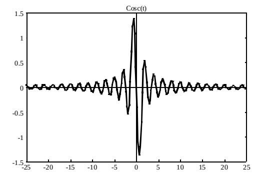
Theory-Math | Theory-SP | Interpolation | **Derivative**

Ideal Derivatives

Spatial Domain:

convolution is exact

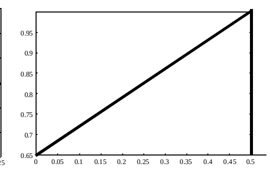
$$f_r^d(x) - f'(x) = 0$$



Frequency Domain:

cut off freq. replica

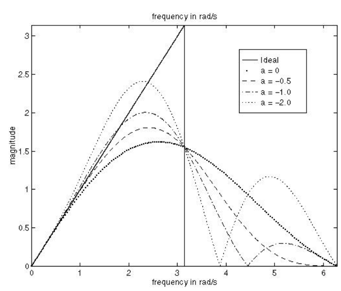
$$\mathrm{Cosc}(x) = \frac{\cos(\pi x)}{x} - \frac{\sin(\pi x)}{\pi x^2}$$



Theory-Math | Theory-SP | Interpolation | **Derivative**

Bentum, Lichtenbelt, Malzbender

- Spatial domain method
- study gradient filter according to analytical derivative of the interpolation filter.



Theory-Math | Theory-SP | Interpolation | **Derivative**

Spatial Domain Evaluation

- Reconstruction of the N^{th} derivative:

$$f_r(t) = a_0 f + \ldots + a_{N-1} f^{(N-1)} + a_N f^{(N)} + a_{N+1} f^{(N+1)} + \ldots$$

Analyze	Are all $a_l(\tau) = 0$ for $l<N$?
Normalize	Is $a_N(\tau) = 1$?
Classify	How many $a_l(\tau)=0$ for $l>N$?
Error	How large is the error term?

Theory-Math | Theory-SP | Interpolation | **Derivative**

Example

- Good interpolation filters
  $a_0(\tau) = 1$ $a_1(\tau) \approx 0$ $a_2(\tau) \approx 0$
- Good derivative filters
  $a_0(\tau) = 0$ $a_1(\tau) = 1$ $a_2(\tau) \approx 0$
- Good 2nd derivative filters
  $a_0(\tau) = 0$ $a_1(\tau) = 0$ $a_2(\tau) = 1$

Theory-Math | Theory-SP | **Interpolation** | Derivative

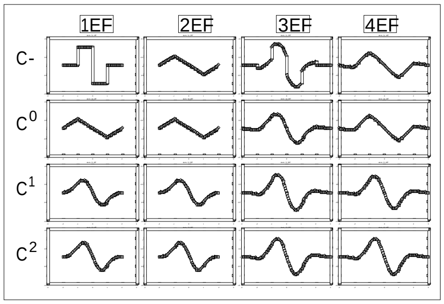
Spatial Filter Design

Which **DERIVATIVE** of the original function do we want to reconstruct?

What **ACCURACY** do we require from the reconstruction process?

How **SMOOTH** (space C^n) should the reconstructed function be?

Derivative Filters



Normal Estimation Schemes

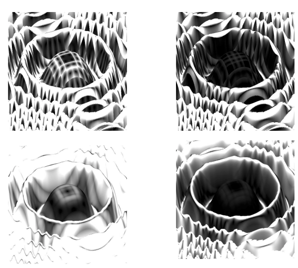
- Method (FD)H - derivative first
  - Compute normal at grid points
  - Interpolate new normals
- Method (FH)D - interpolation first
  - Reconstruct continuous function *f(t)*
  - Apply discrete derivative filter
- Method F(DH) - continuous derivative (new)
  - Convolve derivative filter with interpolation filter
  - Apply this filter to the data samples
- Method FH'- analytic derivative
  - Analytical derivative of the interpolation filter

Results - Volume Rendering

linear error derivative filter | cubic error derivative filter

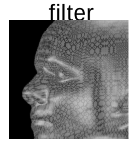
discontinuous derivative filter

C^0 continuous derivative filter

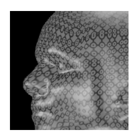


Results - Pattern Mapping

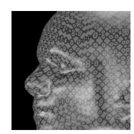
discontinuous linear error derivative filter



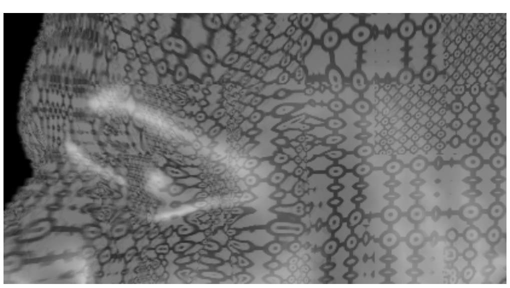
C^0 continuous quadratic error derivative filter



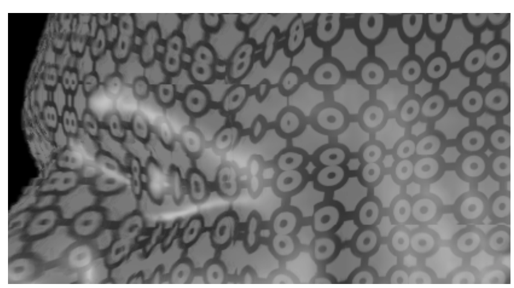
C^1 continuous quadratic error derivative filter



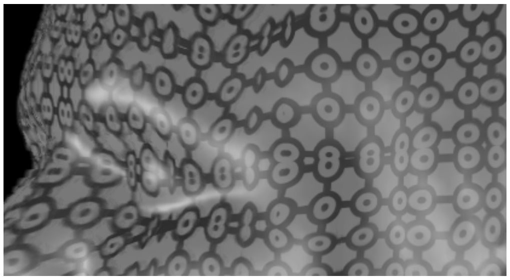
Results - Pattern Mapping



Results - Pattern Mapping



Results - Pattern Mapping



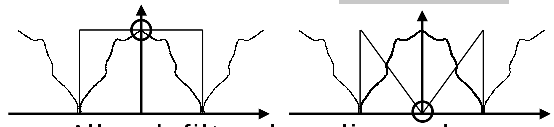
Frequency Considerations

Accuracy

$$H^{(n)}(0) = j^n a_n(\tau)$$

Smoothness

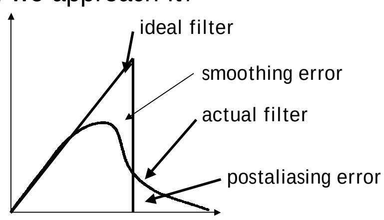
$$|H(\omega)| \leq \frac{C}{|\omega|^M}$$



All such filters have linear phase.

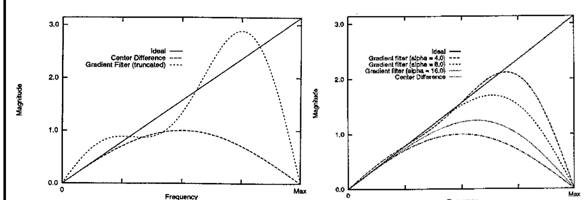
Frequency Domain Design

- We are far from ideal filtering.
- How can we approach it?



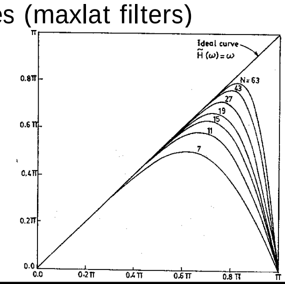
Quick Fix - Windowing (Goss)

- Window of ideal derivative filter (Cosc).
- Kaiser window (based on Bessel Functions)



Dutta Roy & Kumar

- Based on maximal linearity at desired frequencies (maxlat filters)



Summary

- Spatial domain approach allows intuitive constraints on filters
  - (local) accuracy
  - smoothness
- Frequency domain approach allows
  - interpretation of designs
  - wise choice of parameters